Providing robotic assistance during extra-vehicular activity

Robert R. Burridge[*a], Jeffrey Graham[**b]
[a]S&K Electronics; [b]Titan-Lincom

ABSTRACT

Manned missions to other planetary bodies will rely heavily on robotics and automation to enhance the operational safety and capabilities of the crew. In particular, the movement and sensing capabilities of humans in spacesuits are severely constrained. Thus, an important class of robot will be those that accompany humans during extra-vehicular activity (EVA) and provide assistance -- tool transport, video documentation, sample collection, etc. In 1999, NASA engaged in a set of field tests in California called ASRO (AStronaut-ROver), in which a space-suited test subject collaborated with the tele-operated Marsokhod mobile robot, controlled by scientist at a remote location.

From the lessons learned in the ASRO tests, the *EVA Robotic Assistant* project was started at NASA's Johnson Space Center to provide a testbed for continued research in astronaut-robot interaction and cooperation. In September 2000, NASA conducted two weeks of field tests in Arizona at three planetary surface analog sites. Three scenarios were tested requiring cooperation between a space-suited astronaut and the autonomous EVA Robotic Assistant: "Power Cable Deployment", "Solar Panel Deployment", and "Geologist's Assistant". In this paper, we describe the ERA project in detail, and report on results from the Arizona field tests.

Keywords: human-robot interaction, human/robot cooperation, space robotics, robot autonomy, robot assistant, autonomous agents, CORBA, robot control architectures

1. INTRODUCTION

As human missions into space expand in complexity and duration, crewmembers will increasingly rely on automation and robotics in all aspects of their environment and mission[1, 2]. This will be due to three main factors:

1. The scarcity of crewmember time will necessitate the automation of mundane tasks to allow the crew to prioritize objectives and apply their expertise appropriately
2. The volume of assembly, inspection, maintenance, and exploration tasks required for an extended mission will necessitate that some be done without human involvement.
3. The need to minimize the inherent risk associated with extra-vehicular activity (EVA) requires that robots perform some of the more hazardous activities.

Astronauts need substantial assistance when performing extra-vehicular activity. Spacesuits perform two critical functions: protection from radiation and thermal extremes and supplying a breathable atmosphere. Unfortunately, current spacesuit technology produces suits that are bulky, have significant mass, and drastically restrict the astronaut's mobility, dexterity, and visual field. Despite these limitations, many EVA tasks will continue to require the physical and mental involvement of humans at the work site. We wish to understand the nature of these tasks, and how to design robots that can enhance the safety and productivity of astronauts during EVA.

The *EVA Robotic Assistant* (ERA) project at NASA's Johnson Space Center (JSC) was started as a result of a series of field tests in California called ASRO[3], which were a collaboration between JSC and NASA's Ames Research Center (ARC), using a robot called Marsokhod. The ERA team is specifically interested in the issues of how to produce a robot that can assist someone in a spacesuit. Some of these issues include astronaut/robot communication, such as voice or gesture; appropriate size, speed, capacity, sensors, manipulators, processors, and tasks for the robot; instrumentation on the spacesuit or in the habitat that give crewmembers access to the "mind" of the robot; and various levels of autonomy for the robot. This project is a close collaboration among various groups at JSC, including Intelligent Systems, Robotics,

[*] Robert.r.burridge1@jsc.nasa.gov; phone (281) 483-3455; fax (281) 461-9550;website http://www.traclabs.com/~burridge; S&K Electronics, 1016 Hercules Drive, Houston, TX 77058
[**] Jgraham@lincom-asg.com; phone (281) 483-2743; fax: (281) 488-0191; Titan-Lincom, 1020 Bay Area Blvd., Suite 200, Houston, TX 77058

Planetary Science, the Advanced Spacesuit Lab, Exploration Office, and EVA Project Office. We are also involved with Carnegie Mellon University, ARC, the Jet Propulsion Lab (JPL), and Glenn Research Center.

One of the central themes of the ERA project is conducting outdoor field tests at JSC's Planetary Surface Simulator and at remote field sites. In September 2000 we joined the Advanced Spacesuit Lab for two weeks of field tests near Flagstaff, AZ, exploring three scenarios requiring human/robot cooperation.

Section 2 of this paper covers some background for this project, including the ASRO field tests, other current research in human/robot interaction, and adjustable autonomy. Section 3 describes the robot system in detail, both hardware and software. In Section 4, we describe the Flagstaff field tests. In Section 5 we discuss current activity and future directions, and in Section 6 we conclude.

2. BACKGROUND

The ERA project arose after a series of experiments conducted in California in 1999 called ASRO, and incorporates ideas from several subfields of robotics and Artificial Intelligence, such as Human/Robot interaction and adjustable autonomy. These subjects are briefly described in this section.

2.1 ASRO

In February 1999, researchers from NASA's Ames Research Center and Johnson Space Center conducted a series of experiments in astronaut/robot interaction in Silver Lake, California. These experiments, dubbed ASRO (AStronaut-ROver), were some of the first to explicitly examine the issues of using a mobile robot together with a person in a spacesuit. For these experiments, Ames' Marsokhod rover was used, incorporating some stereo vision software developed at Johnson.

During the ASRO experiments, four science scenarios were tested: (1) rover as scout, where the rover is sent into an area to gather data prior to a human traverse; (2) rover as videographer, where the rover provides video coverage of the astronaut; (3) rover as field science assistant, where the astronaut places colored flags at locations of interest and the robot follows, performing tasks at each site according to which flag is present; and (4) rover as field technician assistant, where the rover carries tools and samples for the astronaut[1].

In all of these experiments, the movement of the Marsokhod was controlled via teleoperation from a remote control station approximately 1.5km from the test site. In addition, there was a remote Mission Control Center at Ames (800km from the site) which included a science support team that helped mak e real-time scientific decisions.

One of the most important lessons learned from the ASRO experiments is that the robot must be able to keep pace with the human. The Marsokhod was designed for low energy consumption, and thus was roughly ten times slower than the person it was assisting. This forced the human to take numerous breaks to wait for the robot, wasting time and life-support expendables. Another lesson was that the science support team had great difficulty communicating with the EVA test subject when trying to reference particular rocks of interest. If the robot is to support remote scientists in real-time interaction with the EVA crewmember, then it must have some way to indicate a specific terrain feature[4].

2.2 Human/Robot interaction

There are many researchers investigating various aspects of human/robot interaction. Breazeal et al. at the MIT Media Lab are interested in building "socially intelligent" robots that can interact with and learn from humans in an intuitive manner. These robots, such as Cog and Kismet, tend to be fixed to a location, interacting with people that approach them, but otherwise not interacting with their environment[5].

Aside from face-to-face, another way for humans and robots to interact is for the human to teleoperate the robot. NASA's Robonaut project represents the state of the art in this arena[6]. The Robonaut has 43 degrees-of-freedom in an anthropomorphic upper body, with a waist, two 7-dof arms with five-fingered 12-dof hands, and a head on an articulated neck. The operator uses a Virtual Reality visor to see what the Robonaut's eyes see, and wears gloves and other position sensors that drive the various parts of the Robonaut. Work on automating this robot is just beginning to yield results, but full intelligent automation is still years away.

An interesting type of robot expected to interact with people is the "Museumbot", which is expected to provide museum visitors with information about what they are seeing. Like the social robots, these robots are not expected to physically engage people or the environment, however they are mobile, and must be able to navigate safely around people[7, 8].

The ERA project should benefit from research in this area of robot social intelligence. The very nature of a mixed astronaut-robot team in which they must rely on each other to perform and possibly survive requires a level of social interaction that is intuitive, reliable, and instills trust.

2.3 Adjustable autonomy

In addition to social intelligence, the issues of adjustable autonomy must be addressed for human/robot interaction. EVA scenarios exist that require a mixed team of astronauts and robots. Such scenarios will require the full range of adjustable autonomy-- the ability to dynamically change the level of control and interaction of humans with robotic agents- from extreme autonomy (in which human interaction is impossible or unwanted) to teleoperation, voice and manual control.

The ERA project is exploring the goal of adjustable autonomy-- to design highly capable autonomous systems that are human-centered, i.e., the system design maximizes the goals of the human and supports a full range of interaction[9, 10]. These human-centered autonomous systems will minimize the necessity for human interaction, but maximize the capability for humans to interact at whatever level of control is most appropriate.

3. ROBOT SYSTEM

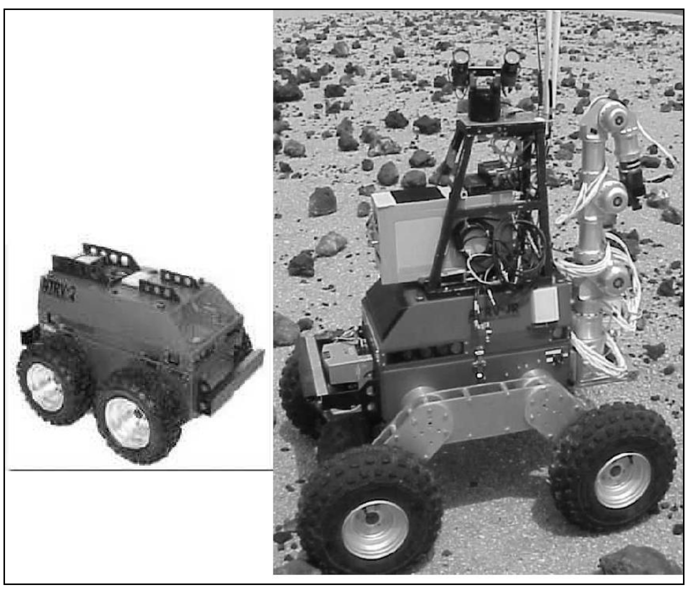

Figure 1:Left: The ATRV-JR as delivered; Right: As modified, including the Metrica manipulator (see Section 5).

The ERA mobile platform is a testbed for experiments in human/robot interaction, particularly when the human is constrained by a spacesuit. Based on the results of the ASRO experiments, a commercial mobile robotic platform was chosen that was faster than a walking human, and large enough to provide a suited crewmember convenient access to tool palettes. Processors and sensors were chosen to support fully autonomous robot activity, with no off-board processing or human intervention. In fact, the large majority of effort on this project has been in developing the software for autonomous robot behavior.

3.1 Hardware

It must be stressed that the EVA Robotic Assistant is not intended to be a flight robot, but rather a research platform. As such, we use technologies that are not currently certified for flight. We are able to do this because human missions to other planetary bodies are so far in the future. If a particular capability is found to be so useful that it enters the critical path for a successful mission, then we can establish a requirement for the supporting technologies to be developed for flight. Indeed, one of our primary objectives is to collect enough data on astronaut/robot interaction to have an impact on the Mars Reference Mission[1], providing design constraints for those who will build the flight system. Thus, the ERA robot uses Pentium CPUs although they are not radiation-hardened, and Differential GPS despite the lack of beacon satellites orbiting Mars (though the Mars Reference Mission does mention a navigation system infrastructure).

The ERA is based upon the ATRV-JR, a commercial mobile platform manufactured by RWI, Inc.[*], delivered with sonar, laser rangefinder, Inertial Measurement Unit (IMU), inclinometers, compass, GPS, speech generator, and two Pentium computers. The robot has subsequently been modified extensively to meet the demands of the typical EVA task and terrain (see Figure 1). In particular,

- The wheel hubs were moved out and down, and larger wheels were installed, increasing the ground clearance from an unusable three inches to twelve inches. This improvement effectively prevents the robot from "high-centering": should one of the new hubs get caught up on a rock, it is far enough away from the center of the robot that the other wheels maintain sufficient traction to drive the robot over the obstacle.
- A pan-tilt-verge active camera platform was installed with a stereo pair of cameras.
- A tower was added for mounting the cameras up almost to eye level, and for mounting various antennas.
- The sonar and original GPS sensors were removed, as neither could provide useful data in the planetary surface analog sites of interest (we are currently installing a differential GPS system that will provide 2cm accuracy).
- A high-fidelity radio receiver was installed and connected to a sound card in one of the computers. This enabled the ERA to receive and interpret voice commands using IBM's ViaVoice software.
- A trailer hitch was added, and a trailer was built for deploying power cables or a flexible solar panel array.
- Tool palettes were built to fit along each side of the robot tower, in easy reach of the astronaut and at such height as to minimize his movements associated with their use. These included a place to store rock samples.

In Section 5 we describe further hardware modifications that are currently under way, aimed at enabling the robot to perform new, more complex tasks.

3.2 Software

The EVA Robotic Assistant (ERA) software is an agent-based architecture built with the goal of producing high quality code that promotes interoperability in all its forms, portability, code reuse and (as the software matures) code sharing among various other robotics groups. These goals are realized using open-source tools that adhere to non-proprietary, recognized standards for interoperability (CORBA), operating systems (POSIX), and programming languages (ANSI and ISO). The following paragraphs describe the tools, control architecture and details about the ERA software and how it interacts with the sensors, actuators, and suited astronauts.

The ERA software map is replete with servers running in parallel across multiple CPUs. Figure 2 shows the existing and near-future baseline (an asterisk "*" indicates under development, a "?" represents possible future direction). Greyed ovals represent lower level servers connected directly to sensors or actuators while white ovals represent higher-level servers. Rectangles represent libraries. Baseline 0 (those pieces of the software map not currently under construction) was used in the field trials discussed in Section 4. Baseline 1 is the current effort.

Of special interest in the software map are the single existence, global servers labeled "Goodness Map", "Location and Pose", and "Path Planner". Their global nature allows multiple robotic agents to update and expand one world map and making the improving path planning available to each other and other sensor-challenged mobile robots. The global path planner will be able to coordinate the activities of multiple human and robotic agents by interfacing with the global location and pose server, which tracks the whereabouts and orientation of any identified objects submitted by agents.

[*] Now a subsidiary of iRobot, Inc.

This is the beginning of support for efficient, mixed-initiative multi-agent coordination in the ERA architecture. This server will include the ability to fuse multi-source or noisy data into a more accurate location and pose using techniques such as Kalman filtering.

The ERA architecture incorporates adjustable autonomy with its behavioral modes and verbal interaction. The architecture will support the collection of metric data for evaluating the performance via its logger server and timeHelper libraries (see discussion below). The Remote Workstation offers a graphical, context-based insight into state of human/robot EVA mission and adjustable autonomy.

3.2.1 CORBA

The ERA software architecture achieves a high degree of portability and interoperability by leveraging the many benefits of the Common Object Request Broker (CORBA). CORBA is the middleware "glue" that binds its clients and servers together and facilitates communications with other software agents.

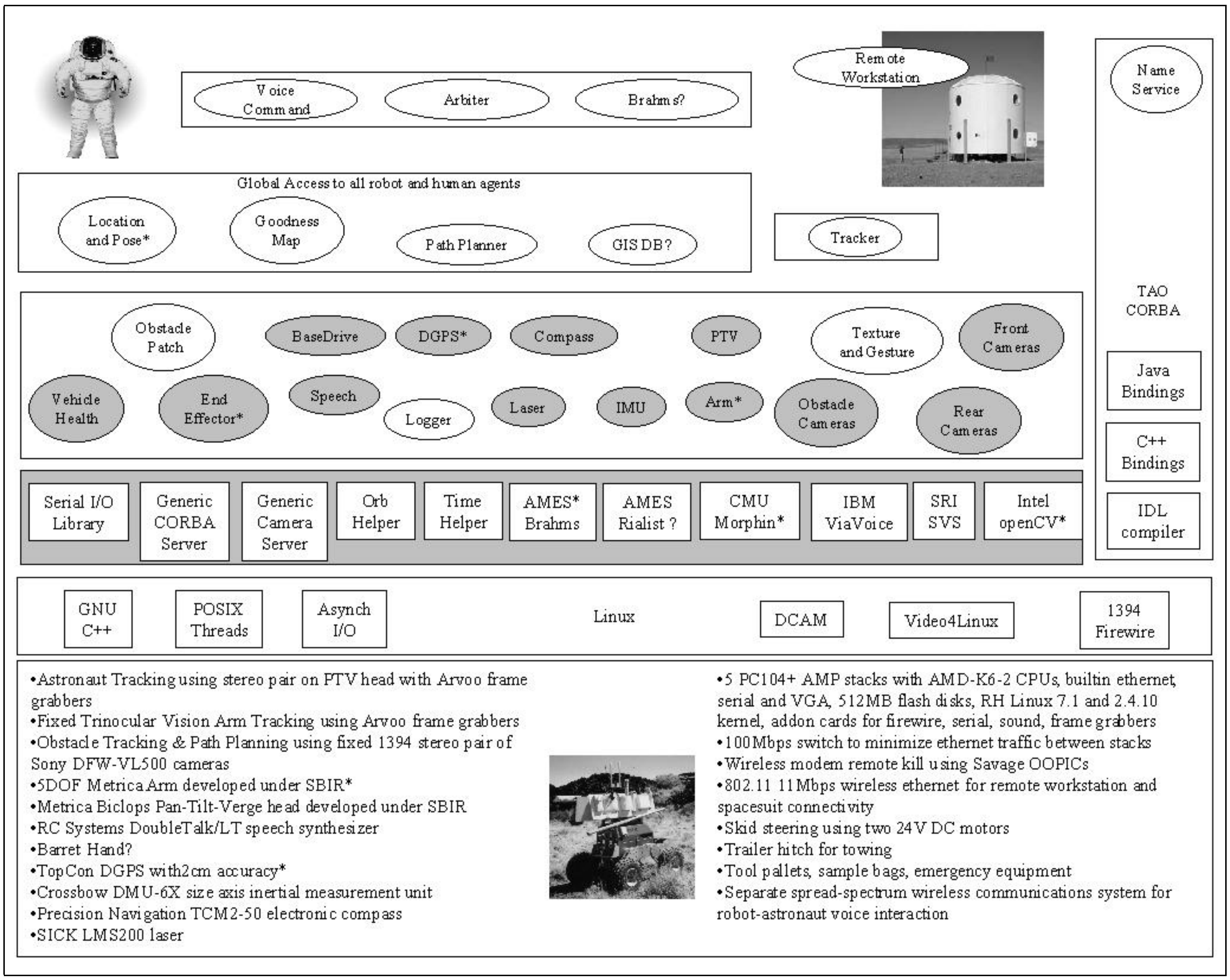

Figure 2: ERA architecture

CORBA is a set of existing standards for software interoperability, as defined by the Object Management Group- a non-profit consortium of over 800 international companies[11, 12]. Popular alternative intra- and inter-process communications infrastructures including IPC[13], NDDS[14], and raw sockets fall short in offering the same functionality and ease-of-use that CORBA provides. Indeed, ERA is currently leveraging the many powerful benefits that CORBA offers[15]:

- international standard for middleware
- hardware independence: endeanness, marshaling and unmarshaling, transport mechanisms
- operating system independence: intermix Linux, Solaris, Windows, VxWorks, etc.
- programming language independence: intermix C, C++, JAVA, LISP, etc. Use the "right language" for the job.
- location independence: servers and clients can run anywhere without change. Some orbs even automatically and transparently optimize inter-process communication (labeled the "collocation optimization" by noticing servers and clients within the same process or on the same CPU and using shared memory or other internal mechanisms to avoid the overhead of a loopback or ethernet device.

CORBA provides an Interface Definition Language (IDL) used to declare typedefs, enums, structures, lists, exceptions, servers and their well-defined public interface. An IDL compiler consumes this file and produces source and header files that map the IDL into the language of choice (C++ and Java for the ERA project). These source and header files are then compiled along with other application files and linked into a client or server application, which treats instances of these CORBA objects like familiar smart pointer "references", free from worrying about the underlying mechanics of the communications.

Because of CORBA and the multiple networked CPU cores onboard the ERA robot, the ERA software executes with a high degree of parallelism. Processes are distributed logically onto the CPUs hosting the required hardware, sensors and actuators. Load balancing is achieved simply by changing where processes execute and requires absolutely no changes in the software itself.

In its next baseline, ERA will benefit from additional CORBA facilities and services, including:

- Real Time CORBA specification and its Quality-of-Service guarantees that manage bandwidth, latency, jitter, and dependability[16, 17]
- Implementation repository service that auto-launches servers on demand
- Notification Service for asynchronous, non-polled data exchange to minimize message overhead
- Trader Service to discover servers that provide a needed functionality at runtime.

3.2.2 Open source

The ERA software is built using open source software for the operating system, CORBA implementation, and development environment.

Linux is an ideal operating system for robotics applications for many reasons. Native support for IEEE 1394 "Firewire" devices, Video-4-Linux support for some popular frame grabbers, and support for the Digital Camera specification make Linux a reasonable choice for vision processing. The ability to customize the Linux kernel by compiling only the essential functionality helps Linux meet the limitations and requirements of a wider range of embedded applications than other operating systems. Finally, familiarity with Linux by the ERA team and its zero-cost make it the operating system of choice for the project. Real-time concerns are not a problem yet since there are sufficient quantity of "fast enough" processors for untethered fully autonomous control.

The primary CORBA implementation for Baseline 0 of the ERA software was OmniORB. For Baseline 1, however, another popular open source implementation called The Ace Orb (TAO) is being used because it offers more of the interesting CORBA services that will benefit the project, more closely tracks the CORBA standards, is aggressively being improved, and runs on most popular operating systems.

Another essential element for the ERA project is its development environment. As is customary for the Linux operating system, the ERA project is utilizing the Gnu tool suite including its C++ compiler, linker, assembler and make utility.

Java and its capability for “write once, run anywhere” graphical user interfaces is being leveraged in that role for the ERA’s Remote Workstation. This tool is capable of displaying video from the several different camera sources, overlay telemetry and other data, and allow the remote user to interact with the robot.

3.2.3 Reuse, Portability and Quality

A framework for reuse, portability, and quality is in place, consisting of thorough object-oriented analysis and design, strict compiles with no warnings, code walkthroughs, libraries of reusable code and class hierarchies. Currently, the framework consists of the serial library, orb abstraction library, time helper, the aforementioned generic server, an abstraction of a video server, and an abstraction of a pan-tilt-verge head.

The primary interface to the sensors and actuators onboard the ERA robot is the venerable serial port. A serial class has been written that supports asynchronous, select-based, and raw I/O. This class also features canonical and non-canonical mode processing. Differences among the various vendor implementations of CORBA have been abstracted away in the orbHelper class. This class manages orb initialization, naming service registration and lookup, signals and graceful shutdown. The TimeHelper class handles temporal functionality for the ERA servers. Time, elapsed time, sleep, and date functionalities are all encapsulated here.

The Generic CORBA Server class forms the basis for all ERA servers, providing the ability to initialize, enable, disable, exit, and query the status of any derived server. The near-future version of the generic server will offer command line parsing, worker thread pool, and configuration file support. The arbiter and its facades (the remote workstation and voice command server) benefit in the form of simplified coding since they can cast/narrow to the base class generic server to interact with the other servers, without bothering with the specialized class header files, class names, etc.

The Generic video server was written to solve the recurring problem that all vision researchers face trying to share code—they spend many man-months reinventing their own low-level service to provide image data to their applications. The generic video server solves this problem in object-oriented fashion by abstracting away and thus hiding the details of image acquisition, while providing a consistent interface (via its IDL) without regard to the underlying hardware. Currently, the IEEE 1394 interface is supported as well as the generic video-4-linux interface. It is probable that an additional interface will be added for the Matrox brand of frame grabber cards. An interesting benefit for the client applications that use this interface is that they can change the hardware source of the raw image data by simply connecting to a different video server (using the Naming service). No code change to client code is necessary! As different low-level frame grabbing devices support different features (such as zoom, onboard clipping, synchronizing grabs, etc.), the generic video server will throw an “unsupported” exception should the client ask for a functionality that is not supported by the named video server it is connected to.

Similarly, the Pan-Tilt-Verge Server hides the low-level details of which pan-tilt-verge head is being used. The different hardware implementations (Metrica’s Biclops and Zebra) register with the CORBA naming service with unique names, allowing clients to change their choice of hardware by changing the name of the PTV and nothing else.

3.2.4 Flexibility, Architectures, Integration

Thanks to the well-defined interfaces via C++ and CORBA, and the multiple independently-executing servers running on multiple CPUs, the ERA software architecture is extremely flexible. An arbiter coordinates resource utilization (servers) and this simple control has worked well thus far. However, as the task complexity increases and thus the need for better coordination and planning also increases, the use of a more formal control architecture will become necessary. The flexibility inherent in the ERA architecture will simplify the adaptation of something more formal such as the three tiered (planning, executive, functional) architectures like 3T, TDL, and others such as CLARAty or Remote Agent.

Superficially, the ERA architecture maps easily onto “3T”, a proven three-tiered architecture in use at NASA/JSC[18]. A possible mapping onto 3T would label ERA's lower-level servers as “functional skills”, its arbiter as the “sequencer/executive” and the addition of a new server wrapping the Adversarial Planner into a server as the top tier goal maker.

CLARAty is a 2-tiered architecture similar to the ERA[19]. Its "functional" Lower layer readily maps onto ERA's servers connected to sensors and actuators, and at the top of the functional pyramid would be ERA's behavioral servers such as Tracking. As in CLARAty, the ERA architecture pushes down and encapsulates decision making to the lowest possible layers. The ERA arbiter could be enhanced to include such task primitives as "move" and "grasp" and thus mark "the line" between the upper "decision layer" and lower "functional layer".

ERA has integrated several outstanding software agents into its architecture to provide higher level control. Integration is nearing completion of the Morphin local obstacle avoidance planning software from Carnegie-Mellon University[20]. IBM's ViaVoice for Linux has been successfully integrated for its speech recognition capability allowing the astronaut to speak to the ERA robot and command it into any of the various behavioral modes. Finally, Metrica's texture tracking software has been successfully integrated as a standalone agent, providing the ERA robot capability to track and follow an astronaut on an EVA traverse[21].

4. FIELD TRIALS

Three representative scenarios for planetary surface operations were developed that required varying degrees of cooperation between the rover and a suited crewmember[22]. In September 2000, the ERA team and JSC's Advanced Spacesuit Lab spent two weeks near Flagstaff, Arizona, conducting joint trials of these scenarios at three planetary surface analog sites.

4.1 Scenarios

Figure 3: power cable deployment (l), solar panel deployment (m), geology traverse (r)

The **power cable deployment** task was motivated by the need to run a significant (> 1km) length of cable from a potential nuclear power source to a habitat[1]. However, for these field tests, the cable was limited to 300ft. The cable reels were mounted on a trailer that was pulled by the robot (see Figure 3-l). The robot tracks the suited astronaut using its stereo vision and/or laser sensors. This task illustrates perfectly the symbiotic advantages of a mixed-team: the cables are too bulky for the astronaut to deploy, and the robot is not capable (currently) of choosing the best path. The two together perform a task neither could accomplish alone.

The **solar panel deployment** task was motivated by the possibility of needing to unreel lightweight flexible solar panels that would provide power to a habitat. The robot must carry the solar panel material because it is too heavy and bulky for the suited astronaut, and must drive in a straight line to avoid kinking the flexible solar panel material (see Figure 3-m). The astronaut interacts with the robot, verbally commanding it into the mode and setting the work pace by telling it to speed up or slow down.

In the **Geology Traverse** scenario, the robot carries geologic tools for an astronaut on a geology traverse, and provides storage space for collected samples. The robot tracks and follows the astronaut using stereo vision, attempting to maintain a fixed separation (see Figure 3-r). The astronaut interacts with the robot by commanding it into the desired mode and telling it to pause and resume when needing to approach the robot for tools or samples .

4.2 Sites

Three sites near Flagstaff were chosen that had been identified by the United States Geologic Survey (USGS) as good planetary surface analogs.

- **Cinder Lake**: A mile-wide volcanic ash bed representative of a lunar maria area. This flat ash bed was bombed so that the Apollo astronauts could train with craters, and many of those craters are still evident.
- **SP Mountain**: A lava flow area representative of a young Martian volcanic feature. The terrain features large piles of loose rocks and steep slopes.
- **Meteor Crater**: The ejecta field just beyond the rim of the crater, representative of Lunar or Martian craters. This area featured steep slopes, but the rocks were not as loose as those at SP Mountain.

In addition to the remote field sites, the ERA project and the Advanced Spacesuit Lab conducted several preliminary tests at Johnson Space Center's 100 foot square Mars surface analogue site, which was completed in May, 2000 [23]. This enabled the teams to identify and fix problems prior to the major field tests.

4.3 Results and Lessons Learned

The suited test subjects and robot were able to successfully accomplish the goals of the various scenarios. We were particularly excited by the fact that the test subjects interacted with the robot as a behavioral agent, i.e., they were clearly building internal models of how it would react to them, and choosing their actions accordingly. Furthermore, the robot acted autonomously much of the time, an improvement over the ASRO experiments where the robot was always teleoperated. Nevertheless, there was a general sense that the robot was not quite robust enough for the challenging environment, and that technological shortcomings hampered the experiments. In the following paragraphs we mention a few areas that need improvement, and then discuss our current activity in those areas in more detail in Section 5.

Communication. The environment in the pressurized spacesuit was so noisy that, even with a high-fidelity radio link, ViaVoice had trouble recognizing spoken commands. We are currently working with technologists at Glenn Research Center and Kennedy Space Center to solve this problem.

Power consumption. Unlike unmanned science rovers that must survive for weeks on solar power, we believe that a human outpost will have an abundant supply of power. Thus, a robot on such a mission need only last long enough to return to base and recharge. Although the ERA rover batteries lasted longer than the expendables in the advanced spacesuits being tested, the batteries had to be swapped out several times per day during the field tests. Because we believe the robot should be performing autonomous tasks between the times it assists humans, we are seeking ways to extend the "stamina" of the robot. These include discussions with researchers at JSC developing fuel cells.

Suspension. The current rigid suspension is not appropriate for rough terrain. Not only does it transfer shocks directly to the sensors and computers on board the robot, it also reduces traction and makes driving more difficult. We are investigating designs to provide four-wheel independent suspension.

Overall capability. In Flagstaff, the robot was only capable of activity in close proximity to a human, and had no ability to manipulate its environment. After the field tests, more complex scenarios were discussed, many of which would require the robot to navigate independently, pick up tools and samples, and generally have a better internal model of the task being performed by the team.

5. CURRENT ACTIVITY AND FUTURE DIRECTIONS

As a result of the Flagstaff field tests, several desired improvements to the robot's core capabilities were identified:

Manipulation. We are adding a 5 degree-of-freedom manipulator to the robot. The manipulator was designed by Metrica Inc. under an SBIR grant for NASA, and has a roll-pitch-pitch-pitch-roll configuration. The end-effector is a parallel jaw gripper made by Eshed.

Navigation. We are adding stereo vision based obstacle avoidance and terrain mapping software derived from CMU's Morphin and D* packages[24]. We are also adding Differential GPS, with a standard error for localization of under 2cm.

Gesture and behavior recognition. We are installing gesture recognition software developed at JSC[25], and working with CMU to develop behavior identification software. These capabilities will allow the robot to interpret deictic references, and to understand what part of the task the astronaut is engaged in.

Together, these capabilities will enable the robot to perform a number of new tasks. These include:

1. Mapping the terrain autonomously, and presenting the map to humans for activity planning
2. Picking up an indicated rock sample and presenting it to the astronaut
3. Returning to a habitat or transport vehicle to retrieve a tool or stow a sample
4. Storing an accurate map of where the astronaut goes
5. Providing continuous video coverage of the astronaut during a traverse

Based on these capabilities, we are currently developing the next round of scenarios for field tests in September 2002.

Aside from the core capabilities mentioned above, the ERA project team is working with the Advanced Spacesuit Lab to explore different interfaces on the spacesuit, such as PDAs, or heads-up displays, and also looking at instrumenting the suit to provide telemetry about vital statistics that the robot can interpret and provide to the astronaut. Thus the robot could inform the astronaut about how far they are from the habitat, how much air is left, and how much time is available before returning to base.

We are investigating the integration of several software packages developed at ARC. These include "RIALIST" natural language understanding software being developed by RIACS[26], and the intelligent agent modeling software "Brahms", which should provide a planning capability currently nonexistent on the ERA robot[27]. Brahms could provide models of cognition and collaboration capable of facilitating effective teamwork between humans, robots, and other software agents.

6. CONCLUSIONS

For two years the ERA project has matured. Starting with an off-the-shelf robot and minimal functionality, the ERA robot has been continuously improved in all aspects. Field tests at analog surface sites exposed weaknesses that are currently being addressed in Baseline 1 software and hardware. The ERA project now has a strong and flexible foundation for large, diverse, distributed teams of intelligent agents, built upon the CORBA standard. There is heavy reuse via object-oriented analysis and design techniques and the benefits are evident in the flexibility of the ERA software architecture. The software is just reaching a level of maturity where it can be shared with interested robotic research groups.

The ERA project is a collaborative effort between numerous branches and divisions at JSC, as well as other NASA and university sites. The ERA robot is uniquely qualified for studying interaction with humans—no other robot is field capable with the size, speed, endurance, and strength needed to support EVA activity.

The collaborative efforts promise to further enhance this unique capability; the ability to collect performance metrics and perform standardized tests for human-robot interaction, improved endurance via more efficient hardware and possibly better power supply with fuel cell technology. soft suspension, new language understanding, and planner capabilities will all be field tested this fiscal year as part of the ERA project.

ACKNOWLEDGEMENTS

This project would not be possible without the support and assistance of Joe Kosmo and Amy Ross of the Advanced Spacesuit Lab, Kevin Watson of the Exploration Office, and Richard Fullerton of the EVA Project Office. We would also like to thank the members of the EVA Robotic Assistant team without whom the Arizona field tests would not have been possible: Ken Baker, Eric Huber, Genevieve Johnson, and David Jochim. Finally, we would like to thank the USGS facility in Flagstaff, who provided a base of operations for us during the field tests.

REFERENCES

1. J. Charles, W. Knudson, R. Cuningham, W. Roy, T. Smith, J. Gruener, M. Duke, S. Hoffman, D. Hamilton, N. Cabrol, "The Mars Surface Reference Mission: A Description of Human and Robotic Surface Activities", Stephen Hoffman, ed. October 1998.
2. B. Cooper, J. O'Donnell, "Robot Associate Study". Presented to Automation, Robotics and Simulation Division, Code ER, NASA Johnson Space Center 07/26/2000. Performed under contract NAS9-98013, DO #0902.
3. R.C. Trevino, J. J. Kosmo, and N. A. Cabrol, "The ASRO Project: Astronaut-Rover Interaction in Planetary Surface Exploration: Science Plan", JSC 39231, February 1998.
4. J. J. Kosmo, R. Trevino, and A. Ross, "Results and Findings of the Astronaut-Rover (ASRO) Remote Field Site Test: Silver Lake, CA: Feb. 22-25, 1999", JSC 39261, March 1999.
5. Breazeal (Ferrell), C., "A Motivational System for Regulating Human-Robot Interaction", in *Proceedings of AAAI98*, Madison, WI, 1998.
6. Robert O. Ambrose, Hal Aldridge, R. Scott Askew, Robert R. Burridge, William Bluethmann, Myron Diftler, Chris Lovchik, Darby Magruder, and Fredrik Rehnmark, "Robonaut: NASA's Space Humanoid", in *IEEE Intelligent Systems*, Vol. 15 No. 4, July/August 2000.
7. T. Willeke, C. Kunz, I. Nourbakhsh, "The history of the Mobot museum robot series: An Evolutionary Study," Proceedings of FLAIRS, Florida, May, 2001.
8. S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Haehnel, G. Lakemeyer, C. Rosenberg, N. Roy, J. Schulte, D. Schulz, and W. Steiner, "Experiences with two deployed interactive tour-guide robots," Proceedings of the International Conference on Field and Service Robotics (FSR'99), Pittsburgh, PA, August, 1999.
9. B. Pell, D. E. Bernard, S. A. Chien, E. Gat, N. Muscettola, P. P. Nayak, M. D. Wagner, and B. C. Williams, "An Autonomous Spacecraft Agent Prototype," Autonomous Robotics Journal **5(1)**, 1998.
10. R. P. Bonasso, D. Kortenkamp, and T. Whitney, "Using a Robot Control Architecture to Automate Space Shuttle Operations," in Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI/IAII97), AAAI Press, Menlo Park, 1997.
11. www.corba.org
12. Object Management Group, "The Common Object Request Broker: Architecture and Specification , 2.3 ed", June 1999
13. http://www-2.cs.cmu.edu/afs/cs/project/TCA/www/ipc/index.html
14. http://www.rti.com/products/ndds/ndds.html
15. M. Henning, S. Vinoski, *Advanced CORBA Programming with C++*, Addison-Wesley, Reading MA. 1999.
16. D. Schmidt, Kuhns, "An Overview of the Real-Time CORBA Specification," IEEE Computer, June 2000
17. Object Management Group, "Realtime CORBA Joint Revised Submission", OMG document, ORBOS/99-02-12 ed., March, 1999
18. Bonasso, R. P., D. Kortenkamp, D. Miller, M. Slack, "Experiences with an architecture for intelligent, reactive agents," Proceedings IJCAI Workshop on Agent Theories, Architectures, and Languages, 1995
19. T. Estlin, R. Volpe, I. Nesnas, D. Mutz, F. Fisher, B. Englehardt, S. Chien, "Decision-Making in a Robotic Architecture for Autonomy," 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Montreal, 2001.
20. http://www.frc.ri.cmu.edu/projects/mars/morphin.html
21. E. Huber, D. Kortenkamp, "Using stereo vision to pursue moving targets with a mobile robot," IEEE International Conference on Robotics and Automation, 1995.
22. J. Kosmo and A. Ross, "Results and Findings of the Representative Planetary Surface EVA Deployment Task Activities, Remote Field Site Test Location: Flagstaff, AZ (Sept. 2-15, 2000)", NASA-JSC CTSD-ADV-470, October, 2000.
23. Lunar/Mars Surface Simulator Field Site Project (Design Report), NASA JSC contract NAS9-19507, 16 June 1998.
24. Anthony Stentz, "The focused D* algorithm for real-time replanning," International Joint Conference on Artificial Intelligence, Montreal, Canada, August 1995

25. D. Kortenkamp, E. Huber, R. Bonnaso, “Recognizing and interpreting gestures on a mobile robot,” Proceedings Thirteenth National Conference on Artificial Intelligence, AAAI, 1996.
26. F. James, M. Rayner, and B.A. Hockey, “Do that again: Evaluating spoken dialogue interfaces,” RIACS Technical Report #00.06, 2000
27. M. Sierhuis, W. Clancey, W. van Hoof, R. de Hoog, “Modeling and simulating Human Activity,” Autonomous Agents 2000 workshop on Intelligent Agents for Computer Supported co-operative Work: Technology and Risks, Barcelona, 2000.